NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A SIMULATION OF READINESS-BASED SPARING POLICIES

by

John D. Wray

June 2017

Thesis Advisor:	Arnold Buss
Second Reader:	Karla Lizzo

This thesis was performed at the MOVES Institute.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2017	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE A SIMULATION OF READINESS-BASED SPARING POLICIES			**5. FUNDING NUMBERS**
6. AUTHOR(S) John D. Wray			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

We develop a simulation to complement a new optimization tool that establishes inventory levels for aviation weapon systems (WS) in the U.S. Navy. The optimization seeks cost minimization while achieving required readiness rates for hundreds of WS, each comprising thousands of indentured parts. Based on work in similar realms, the optimization employs the Vari-Metric model and a variant of a greedy heuristic algorithm to set stock levels and estimate overall WS availability. Our discrete event simulation is then used to test the assumptions of the new optimization tool, compare its performance to other optimization tools available, and provide additional metrics for decision makers. In testing the new optimization tool, we find that (a) there is no systemic bias in estimated readiness; and (b) 53 of 64 WS simulated yield results within 5% difference, with a worst-case difference of 8%. We also test two legacy optimization tools currently in use by the Navy and find they have a larger difference in expected readiness. Finally, we demonstrate additional insights and metrics from the simulation that are not available in the optimization tools.

14. SUBJECT TERMS readiness-based sparing, discrete event simulation, optimization, multi-indenture, inventory, supply			**15. NUMBER OF PAGES** 71
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

A SIMULATION OF READINESS-BASED SPARING POLICIES

John D. Wray
Lieutenant Colonel, United States Marine Corps
B.S., United States Naval Academy, 2000

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN
MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION

from the

NAVAL POSTGRADUATE SCHOOL
June 2017

Approved by: Arnold Buss
Thesis Advisor

Karla Lizzo
Naval Supply Systems Command
Second Reader

Chris Darken
Chair, MOVES Academic Committee

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

We develop a simulation to complement a new optimization tool that establishes inventory levels for aviation weapon systems (WS) in the U.S. Navy. The optimization seeks cost minimization while achieving required readiness rates for hundreds of WS, each comprising thousands of indentured parts. Based on work in similar realms, the optimization employs the Vari-Metric model and a variant of a greedy heuristic algorithm to set stock levels and estimate overall WS availability. Our discrete event simulation is then used to test the assumptions of the new optimization tool, compare its performance to other optimization tools available, and provide additional metrics for decision makers. In testing the new optimization tool, we find that (a) there is no systemic bias in estimated readiness; and (b) 53 of 64 WS simulated yield results within 5% difference, with a worst-case difference of 8%. We also test two legacy optimization tools currently in use by the Navy and find they have a larger difference in expected readiness. Finally, we demonstrate additional insights and metrics from the simulation that are not available in the optimization tools.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

A_o	operational availability
ARROWS	Aviation Readiness Requirements Oriented to Weapons Replaceable Assemblies
DES	discrete event simulations
EBO	expected backorders
FIFO	first in first out
MDT	mean down time
MLDT	mean logistics delay time
MTBF	mean time between failures
MTTR	mean time to repair
NAVARM	Navy Aviation Readiness-Based Sparing Model
NAVSUP	Naval Supply Systems Command
NPS	Naval Postgraduate School
OR	operations research
QPA	quantity per application
RBS	Readiness-Based Sparing
RIMAIR	Retail Inventory Model for Aviation
RSIM	Readiness-Based Sparing Simulation
SPO	Service Planning Optimization
SRA	shop replaceable assemblies
VM	Vari-metric
WRA	weapon replaceable assemblies
WS	weapon system(s)

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

To maintain required combat power for the combatant commanders of the United States naval forces, specified material readiness (i.e., availability) levels must be maintained for all naval aviation Weapon Systems (WS). The term WS here identifies platforms such as the F/A-18 (Hornet) attack aircraft, or the MH60 (Seahawk) helicopter, among others. While reliability and maintainability are primarily set in the design phase of a WS, supportability is a crucial aspect of readiness that can be adjusted throughout the lifecycle of the system to achieve desired readiness rates. Supportability is affected by several factors; one of the key controllable elements is stock levels for spare parts at different echelons of supply.

The United States Chief of Naval Operations requires the use of readiness-based sparing (RBS) in setting inventory levels for most parts. In order to assist Naval Supply Systems Command (NAVSUP) with RBS planning, we developed an RBS Simulation (RSIM) to verify the recently developed Navy Aviation RBS Model (NAVARM) estimates and also compare its performance to the legacy Service Planning Optimization (SPO) and Aviation Readiness Requirements Oriented to Weapons Replaceable Assemblies (ARROWS) tools.

RSIM is a discrete event simulation implemented in the Java programming language using the Simkit library developed at the Naval Postgraduate School. RSIM simulates failures at the individual part level and tracks the impact of these failures on the associated WS. When part failures occur, the part is removed from the system for a set amount of time to simulate repair and ship times. RSIM collects numerous metrics at the WS, part and part position level to provide insights into various factors affecting readiness.

We have run RSIM on seven representative naval sites to compare expected WS availability rates for a given allowancing to those anticipated by NAVARM. The number of WS types at these naval sites ranges from 3 to 23 with a mean of approximately 9 WS types and 111 individual WS (i.e., individual aircraft) per site. Out of the 64 WS types

analyzed, 53 have expected readiness levels within 5% and the mean difference for all WS types in this sample is 0.2% with no systemic bias to over or underestimate readiness noted.

As NAVSUP considers whether to switch RBS optimization tools, it is crucial for the decision makers to assess the accuracy of the NAVARM expected readiness calculations and compare its accuracy to the SPO and ARROWS tools currently in use. Because RSIM models the system at the part and WS level, its method of observing readiness rates through the course of a simulation provides an independent observation to compare against the optimization tool estimates available. SPO, ARROWS, and NAVARM have been run at a representative site with seven different WS types and a total of 62 WS. Their recommended inventory policies have been simulated in RSIM in order to compare the expected readiness rates for each WS type. For all seven WS types at this site, NAVARM's estimates are closer to RSIM observations than SPO or ARROWS estimates.

In addition to verifying NAVARM outputs and providing an independent comparison of the three available RBS optimization tools, RSIM provides additional insights not available with the optimization output. One example of this is the readiness levels. While the optimization tools only provide the average expected levels achieved over a one quarter period, RSIM provides metrics that include percentage of time above the stated readiness goal and the readiness levels observed at the beginning of each simulated day. These metrics can help a decision maker who may be more interested in worst-case scenarios to ensure that assumptions made for contingency planning are realistic.

ACKNOWLEDGMENTS

First I would like to thank my wife for her consistent patience and support throughout my career. Thank you for your understanding when “a couple more minutes” turns into an hour!

To Dr. Buss, thank you for your hard work throughout the process. Your ability to keep me pointed in the right direction and maintain the right scope for the simulation were invaluable. Thank you for all of the assistance and feedback throughout the development process.

To Dr. Salmeron, thank you for guiding me to this great thesis topic and then providing responsive support throughout the development and writing. Your attention to detail and keen understanding of the problem set were crucial in the success of this project.

To Ms. Karla Lizzo, thank you for your constant willingness to help me learn about the supply system and better understand the problem set. You truly exceeded my expectations at every turn and it was great to have a representative from NAVSUP so actively engaged throughout the process.

THIS PAGE INTENTIONALLY LEFT BLANK

THESIS DISCLAIMER

The reader is cautioned that the computer programs developed for this research may not have been exercised for all cases of interest. While every effort has been made with the time available to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

To maintain required combat power for the combatant commanders, specified readiness levels must be maintained for all naval aviation weapon systems (WS). The term WS here identifies platforms such as the F/A-18 (Hornet) attack aircraft, or the MH60 (Seahawk) helicopter, among others. While reliability and maintainability are primarily set in the design phase of a WS, supportability is a crucial aspect of readiness that can be adjusted throughout the life cycle of the system to achieve desired readiness rates. Supportability is affected by several factors; one of the key controllable elements is stock levels for spare parts at a single echelon of supply.

Selecting the right mixtures of parts to stock at any given site is a very challenging task in a budget-constrained environment. A naval site contains numerous WS of different types and each WS may contain thousands of parts, each failing at a different rate. While it may not be possible to identify a provable optimal inventory for every site, our goal is to design and implement optimization and simulation tools that approximate such solutions and provide inventory policies that result in significant cost savings and improved fleet readiness over alternative solutions.

The Naval Supply Systems Command (NAVSUP) is considering replacing their current readiness-based sparing (RBS) tools, Service Planning Optimization (SPO) and Aviation Readiness Requirements Oriented to Weapons (ARROWS), with the Naval Aviation RBS Model (NAVARM) developed by the Naval Postgraduate School (NPS) Operations Research (OR) faculty. NAVARM is designed to inform inventory policy at the site level and is currently undergoing testing at NAVSUP (Salmeron 2016).

NAVARM is designed to provide inventory levels expected to yield required readiness levels at a minimum cost. For this thesis, we use readiness, availability and A_o interchangeably to refer to the fraction of WS that are full mission-capable. Although readiness rates for a given set of inventory allowance levels cannot be calculated with a closed-form formula, NAVARM calculates expected availability using a series of formulas that estimate the expected backorders (EBO) and ultimately the expected

availability of each WS. While these formulas are useful for attaining rapid estimates of availability to compare against policy requirements for availability, their accuracy is unknown.

A. CONTRIBUTIONS

For this thesis we develop a discrete event simulation, the RBS Simulation (RSIM), to gain insights into NAVARM and provide decision makers with additional insights about expected performance. This thesis provides the following contributions:

- An assessment of the accuracy of readiness estimates used by NAVARM—including a range of possible outcomes using stochastic inputs and a comparison of results to NAVARM estimates of expected readiness by WS type.
- A comparison of NAVARM to the two RBS optimization tools currently used by NAVSUP.
- A range of metrics to give a more thorough understanding of what to expect if the NAVARM inventory levels are utilized.

B. SCOPE

There are many details regarding WS readiness that could be simulated to enhance understanding of all factors affecting readiness (e.g., flight operations, shipment process, depot repair process, etc.). While this broad aperture would provide helpful insights, it comes at a significant cost. To model all aspects of the supply system, the coding would be laborious, the run-time would be undesirable and the user interface and data requirements would be complex. Instead, RSIM is tightly scoped to estimate readiness levels given a set of inventory levels, but remains flexible enough to add functionality as required to answer additional questions that may arise in the future.

C. THESIS STRUCTURE

The following is a brief description of the remaining chapters:

- Chapter II reviews applicable research underlying NAVARM.
- Chapter III provides an overview of simulation and describes the RSIM model along with its capabilities and limitations.

- Chapter IV provides analysis of numerous simulation runs resulting in the contributions listed above.
- Chapter V contains summarized conclusions and recommendations for future research in this area.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

In this chapter, we provide a review of RBS and examine tools currently used by NAVSUP to implement RBS. Additionally, we provide an overview of NAVARM, a new tool currently being tested by NAVSUP that will potentially replace the current tools.

A. READINESS-BASED SPARING

The United States Chief of Naval Operations requires the use of RBS in setting inventory levels for most naval aviation parts. Although fill rate is a popular choice for evaluating inventory policies, it is problematic in a military setting where the ultimate goal is sufficient availability of WS (Shebrooke 2004). While improving fill rates or reducing backorders will in fact improve readiness, policies developed with these metrics alone will be inefficient (Moulder et al. 2011). One of the biggest issues with these metrics is they fail to consider the time a part will be on backorder. While this may be nominal in some cases, it is not uncommon to have specialized parts on backorder for over a year; this needs to be factored into an optimal inventory policy. Additionally, looking solely at fill rates will inadvertently punish more complex WS. With all other factors such as failure rates and mean time to repair (MTTR) being equal, a WS with more parts will be requesting more parts from supply. If 95% of the parts are available upon request, a WS with more parts will be unavailable more often while awaiting parts than a WS with fewer parts.

OPNAVINST 3000.12A states that Operational Availability (A_o) “is a primary measure of readiness for WS and equipment. It is determined by reliability (mean time between failures (MTBF)), MTTR, and supportability (Mean Logistics Delay Time (MLDT))” (Chief of Naval Operations 2003, 2). This publication defines A_o as follows:

$$A_o = \frac{MTBF}{MTBF + MDT}, \tag{2.1}$$

where MDT is the mean down time, defined as the sum of MLDT and MTTR. While MTBF and MTTR are dependent on WS and part attributes, MLDT can be significantly reduced by optimizing spare part allocations.

OPNAVINST 4442.5A (Chief of Naval Operations 2011) dictates the use of RBS to achieve required A_o rates. The Chief of Naval Operations is tasked with setting A_o thresholds for all WS types in the inventory. RBS techniques are employed to meet the given thresholds while minimizing cost of spare parts maintained in inventory at the site and enterprise level.

In his textbook *Optimal Inventory Modeling of Systems: Multi-Echelon Techniques Second Edition*, Sherbrooke establishes a process he refers to as the vari-metric (VM) model to conduct RBS optimization (Sherbrooke 2004, 101–125). In a NAVSUP brief to NPS, Cardillo (2016) summarizes the RBS process as follows:

- Calculate the pipeline—"units of an item in repair at a site or being resupplied to the site from a higher echelon" (Sherbrooke 2004, 15);
- Compute EBO—average number of parts on WS that are awaiting supply;
- Calculate expected time the WS is down for a given part failure;
- Develop a ratio of cost per part to downtime expected due to that part failing;
- Rank decisions by cost effectiveness; and
- Select parts in order of cost effectiveness until goal is met.

While this may seem rather simplistic, there are several factors that make this problem very challenging and prevent the use of exact optimization techniques.

B. NAVSUP SYSTEMS USED FOR RBS

Due to the complexity of RBS, the Navy employs a suite of approved tools to set and evaluate inventory levels and policies for RBS at various echelons. This section summarizes capabilities of existing RBS optimization tools available to NAVSUP.

1. ARROWS

ARROWS is a site-level stockage model designed to optimize inventory of aviation parts in multi-indenture structures and seeks to achieve a given availability constraint with minimum cost. The model allows for multiple WS types at a single site and considers the impact of common parts on readiness across platforms. The ARROWS solution to part commonality is rather rudimentary in that it simply optimizes one WS type at a time and considers previously set inventory levels as it moves on to subsequent WS types. To provide additional user input and reduce undesired behaviors, individual items can be constrained to require an absolute stock level, minimum stock level, or maximum stock level to provide some protections or reduce "churn" (the magnitude of change from current inventory levels).

A test of ARROWS availability estimate accuracy was conducted by the Operations Analysis Department of the Navy Fleet Material Support Office using the 1986 deployment of the Enterprise carrier as a data source. The F-14 and SH-60 readiness rates were within 10% of the actual readiness rates reported during this deployment and closely mirrored results produced by a Center for Naval Analysis simulation of the same scenario (Strauch 1986).

2. SPO

SPO is a commercial product developed by Morris Cohen Associates and used in both Department of Defense and industry to optimize inventory at various echelons. SPO is "endorsed and approved to replace ARROWS … ARROWS is already phased out for deployed aviation sites and will be phased out for shore sites in the near term" (Chief of Naval Operations 2011, 13). The methodology for SPO is largely unknown due to its licensing agreements.

3. RIMAIR

The Retail Inventory Model for Aviation (RIMAIR) is included as part of ARROWS and "may be used in provisioning when data is inadequate for RBS modeling or the application of RBS approaches is not cost-effective" (Chief of Naval Operations 2011, 8). NAVSUP uses RIMAIR to provide an "85% Poisson protection level" for aviation ground support equipment, engines and other specifically identified parts. That is, the allowance is set to the 85th percentile of the Poisson distribution. "Its application to low cost items eliminates trade-offs between high cost items with scrubbed data and low cost items whose shear number prohibits effective data scrubbing. Even when optimized separately, however, the impact of these items on availability is computed" (Strauch 1986, B-1).

4. D-SCORE

The Defense Sustainment Chain Operational Readiness Evaluator (D-SCORE) was built for the Department of Defense (DOD) by Logistics Management Institute to examine the impacts of policies, processes or parts on cost and readiness for WS. D-SCORE is a refinement of the Navy Supply Chain Operation Readiness Evaluator and is one of four modules embedded in the WS Sustainment Value Stream Model. D-SCORE was written in SIMSCRIPT 11.5 and is compatible with Microsoft Windows machines up to the XP operating system (Logistics Management Institute, 2008).

D-SCORE is a mature simulation that has been used to answer a number of broad ranging questions across the spectrum of decisions that can influence WS availability. It is capable of simulating up to four echelons of repair (operational, installation, regional and depot), four levels of indenture and up to 20 different types of WS. The simulation has a stochastic demand signal, but all other inputs are deterministic to help isolate the cause of differences and reduce the number of runs required. At varying levels of depth, D-SCORE models transportation of parts between sites, repair prioritization and scheduling, lateral resupply and cannibalization. The broad range of variables that can be adjusted by the user enable analysis of alternative supply policies in a number of different areas with a standardized set of metrics and inputs to the system.

D-SCORE analysis is generally designed to monitor repairable parts in the system. User-inputted usage data (hours flown per day), parts inventory and repair capacity are used to assess failures for each type of WS and move parts through the assessment and repair process. Repair times are based on availability of subcomponents required based on assessed failures as well as the availability of workers at the appropriate echelon of repair.

While the D-SCORE simulation is an effective tool for analyzing a broad range of supply policy decisions, there are several advantages to developing a simulation in-house at NPS. The first advantage is the ability to tailor the simulation to the exact problem being analyzed. With the flexibility of D-SCORE comes a lot of overhead in learning the system and creating the input files required. An open source simulation created at NPS can be modified as necessary to add fidelity in the specific areas of interest and continue to adapt as the analysis requirements change. Additionally, by programming the simulation in Java-based Simkit, developed at NPS, RSIM is offered free of charge and is available and compatible with future operating systems. D-SCORE requires a SIMSCRIPT license and a computer running the Windows XP operating system, which is no longer supported by Microsoft.

C. NAVARM

NAVARM was developed in 2016 in response to a NAVSUP request to identify better allowancing levels for parts in their inventory that have a direct impact on WS availability. NAVARM embeds a heuristic algorithm that approximates the optimal inventory quantities for a single-site, multi-indenture problem. Specifically, NAVARM recommends reorder points that minimize the cost of inventory held while maintaining pre-specified target availability rates for all WS. NAVARM users are also afforded the opportunity to provide minimum and maximum stock levels as well as a starting solution. In addition, some dashboard controls allow for the tool to explore more or less solutions for optimality.

1. Underlying Theory

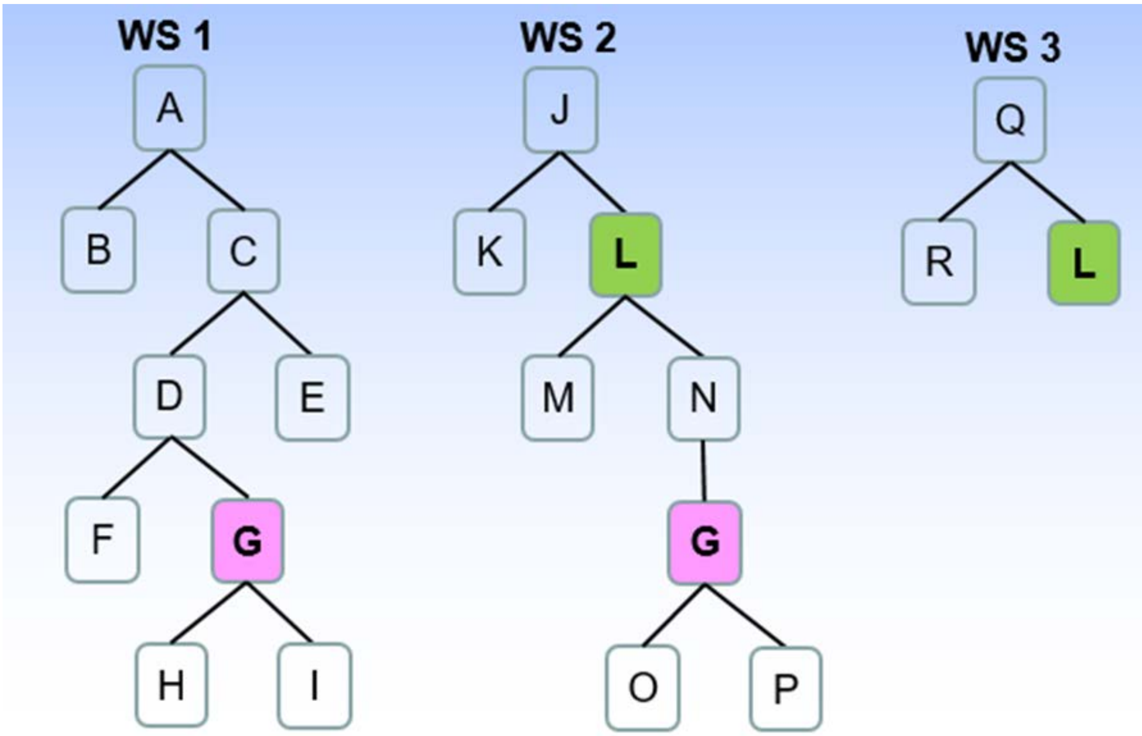
NAVARM assumes an (S-1, S) inventory model for all parts and sites. That is, S is the (maximum) stock level at a site determined by NAVARM and an order is placed as soon as that level decreases by one (i.e., the reorder point is S-1). S is the number on-hand plus the number due-in minus the number of backorders. This means that each time a part fails it is turned into the system for repair. If the part cannot be repaired, a new part is ordered to resupply it. The expected times for repair or resupply are given in the available databases and are currently modeled deterministically in RSIM.

Assuming every part i is given a stock level S_i, every WS has an estimated availability that is calculated as a function of the EBO of the highest indenture parts in the WS. Naturally, backorders for any part in the system are a random variable which depends on: (a) the part's stock level; (b) its (possibly different) failure probability distributions for all common parts in the same or different WS; and (c) the backorder distribution for sub-indentured parts to all common parts. The underlying theory to calculate EBO for a given set of inventory levels S_i follows the abovementioned VM model, see Sherbrooke (2004, 101–125).

The VM model estimates EBO under the assumption that, even though the number of failures for a given part can be modeled using a Poisson distribution, the actual number of failures after accounting for sub-indentured parts' failures follows a Negative Binomial distribution.

The multi-indenture structure used to describe WS repair with more fidelity complicates the problem significantly. The WS itself is treated as the root of the tree and is composed of one to many weapon replaceable assemblies (WRA). In other domains, these WRA are also referred to as line replaceable units because they can be swapped out quickly on the flightline. The next level down the tree consists of shop replaceable assemblies (SRA) also referred to as a shop replaceable unit. Each WRA is composed of zero to many SRAs. Continuing down the tree, these SRAs may be composed of sub-SRA parts. Figure 1 is a graphical depiction of the multi-indenture structure. For NAVSUP purposes, WS may be considered with up to five levels of indenture. The result

of the multi-indenture structure combined with common parts is that a WS of one type may have its readiness affected by the availability of parts in another WS that are not even present in the WS of interest. This is demonstrated in Figure 1.



In the diagram above, if we are interested in improving the availability of WS 3, we can look at ways to decrease backorders of sub-parts "R" and "L." But, noting that part "L" is common to WS 2, its backorders are impacted by parts "M" and "N," and therefore by "G" in WS 2. Moreover, since this is common to WS 1, stocks of parts "H" and "I" in WS 1 will affect backorders of "L" in WS 3. The fact that WS 3 can be influenced by WS 1's parts (which have no direct commonality with parts in WS 3) is a challenging aspect of RBS optimization.

Figure 1. The Chain of Influence in a Multi-indenture Part Structure. Source: Salmeron (2016).

Sherbrooke points out that while the multi-indenture structure and the likelihood of common parts across WS types "does complicate the computer programs substantially … the basic logic is the same" (Sherbrooke 2004, 114).

The use of heuristics to approximate the problem of satisfying a certain availability at minimum cost is justified due to the lack of a closed-form expression for expected readiness rates for a given set of inventory allowance levels.

2. Methodology

To implement the VM model, NAVARM must perform several tasks involving work with the data supplied by NAVSUP in the form of so-called "candidate" files. First NAVARM pulls in data specific to each part position including the part type, indenture level, failure rate, repair time, and ship time. Next NAVARM reads in data specific to each WS type including readiness goal, MTTR, expected flying hours, and number of WS for each type. NAVARM then uses this information to determine indenture levels for each part and systematically works up this hierarchical structure to find common parts across all WS and calculate expected pipelines by part type. This information is ultimately used to calculate the expected readiness by WS type.

The VM model suggests using a greedy heuristic based on an "effectiveness ratio" that measures improvement in EBO with respect to cost. Parts with higher ratios are chosen until the desired availability is met. The use of ratios is predicated on the idea that "EBO decrease" per unit of cost for an additional part is the main driver in the actual optimal decision. While it is easy to build counterexamples where this greedy heuristic would not achieve optimality, it appears to work well in practice.

The matter becomes more complex when there are multiple WS with common parts. This is because if we follow the greedy algorithm for one WS at a time, we will achieve the desired availability at (approximately) minimum cost for, say, WS 1. But then we will need other parts when optimizing part allocations for WS 2. If some of those parts are common to WS 1, we will increase its availability unnecessarily above its target. To help overcome this effect, NAVARM will try optimizing WS types in different orders to see which ordering identifies the most optimal solution. Additionally, NAVARM will perform a number of polishing passes to try reducing inventory for parts on WS types with higher availability than required to find lower cost solutions that still meet all constraints. Both the number of orderings and the number of polishing passes can be set by the user to allow for acceptable run-times and satisfactory results.

3. Assumptions

To perform the optimization, NAVARM makes a number of assumptions. Many of these assumptions are known not to hold in some cases, but are required due to limitations in available data or the desire to simplify the calculations to improve run-time and coding complexity. The following are some of the key assumptions made:

- The NAVSUP supplied formula to estimate average WS availability based on WRA EBO and database inputs is an accurate estimate of mean availability.
- The Negative Binomial distribution is an accurate way to estimate expected failures at all levels of indenture except at the lowest levels of indenture where the Poisson distribution is used.
- The VM model for calculating EBO is correct.
- All part failures result in WS non-availability—partial mission capable WS are not counted as available.
- Parts will not be transferred between sites to cross-level inventory.
- Parts will not be cannibalized from down WS to eliminate backorders for a WS that could be repaired.

D. SIMULATION UTILITY

Comparing recommendations from different RBS tools to determine which tool best meets NAVSUP requirements is difficult. The assumptions and methodology for each tool are distinct and the readiness estimates produced by each tool use different formulas. A simulation of each solution provides an independent estimate of expected readiness and allows for a fair comparison of available tools.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SIMULATION DEVELOPMENT

As detailed in Chapter II, NAVARM relies on a number of assumptions that may not always accurately reflect the results of actual maintenance and supply practices. Simulation provides a tool to help assess the validity of these assumptions. Concepts such as indenture and WS availability that result in significant complexity for NAVARM and other RBS optimization models are straightforward to simulate. In this chapter, we detail the simulation architecture, tools and model used to develop RSIM. Additionally, we specify the output provided and modeling assumptions made.

A. DISCRETE EVENT SIMULATION

There are two main classes of simulation: time step and discrete event simulations (DES). The differences between the two methods is in how time is advanced through the course of the simulation. Results can be significantly impacted by which method is employed. In a time step simulation the state of every entity in the simulation is updated at every time step, the duration of which is set by the user. Unfortunately, "the size of the time step can have a substantial impact on estimated measures of performance" (Buss and Rowaei 2010, 1468). If the time step is too large, events will occur out of order and at the wrong time. If the time step is small, the results will be closer to correct, but significant computing power will be expended checking each entity for updates when most of them will not have changed. By contrast, a DES advances simulation time at rates based on scheduled events to ensure each event happens at the exact time scheduled and in the correct order. While this adds some complexity to the code, it can save significant computational time during a simulation run and will likely improve the accuracy.

The complexity and magnitude of parts flow at the site level is best simulated with a DES. By using a DES, RSIM eliminates the need to check if each part (well over 100,000) has failed or completed repair. Additionally, it ensures that each failure and repair will be simulated at precisely the expected time instead of waiting until the next time step occurs. While employment of DES requires maintenance of a very lengthy event list, a proficiently coded list can be managed efficiently during run time.

The following is a description of key components and concepts related to DES:

1. Entities

Entities in a simulation are objects with attributes that may change through the course of the simulation; thus, they act as a container for variables that we wish to track. Entities may move through the simulation in a manner that allows them to interact with other entities. In the supply realm, entities could be a WS, a part, etc.

2. States

The state space of a simulation is a complete description of all required information to describe the current status in a simulation; it is composed of the values for all state variables. "A state variable in a DES model is one that has a possibility of changing value at least once during any given simulation run" (Buss 2011, 1-1). In a DES, these state variable changes must occur instantaneously and at distinct times; they cannot be continuously changing variables. A state trajectory is a description of the variable's value over time. In the context of simulating maintenance at a given site, state variables would include WS status (available or down for repair), part status (functioning or down for repair), etc.

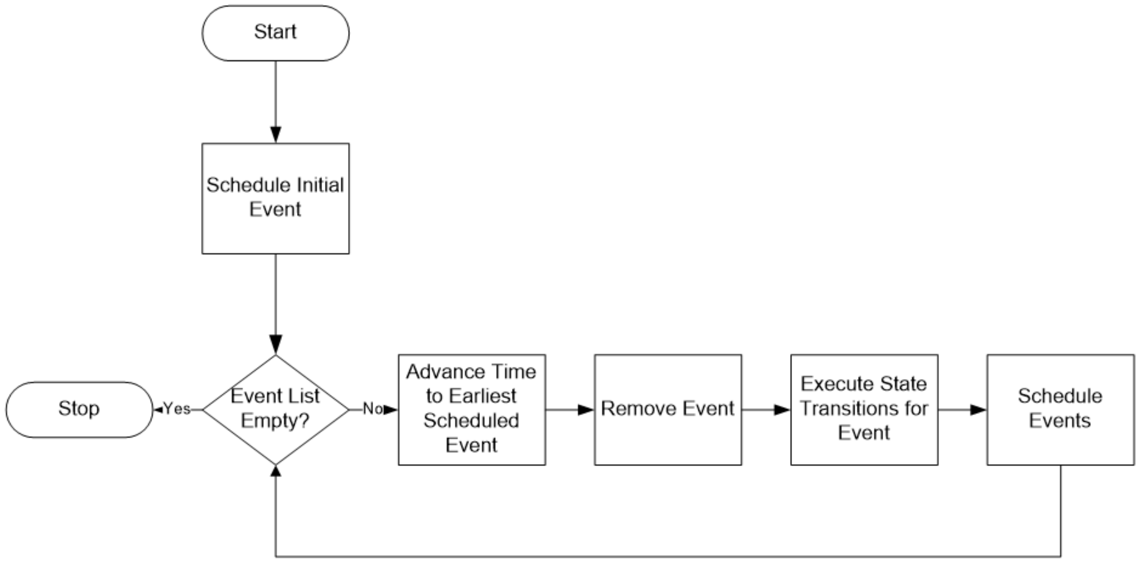
3. Events

"Events are the building blocks of a DES model" (Buss 2011, 1-2); they are used to change state variables and/or schedule other events. Each possible state transition (e.g., a part status changing from functioning to non-functioning) must map to an event that can trigger it. In most cases, an event occurrence will also schedule another event to occur; for example, a part failure event may schedule a part repair event.

4. Scheduling and Time Advance

As previously stated, DES time advance is dictated by events scheduled rather than fixed time intervals defined by the user. As an event occurs, it likely schedules additional events to occur either immediately or at a specified time in the future. An event list maintains all future scheduled events and determines when the time can advance and

to what point. This process continues until the event list is empty or the simulation reaches a user defined stop point. This process is depicted in Figure 2.



This figure depicts the flow of time in a simulation and the crucial role the event list plays in managing the simulation clock at run time. An initial event must be scheduled to initiate the simulation and begin scheduling subsequent events. From here, a time sorted event list is maintained to ensure that all events scheduled to occur at the current simulation time are executed prior to advancing the time clock. The simulation time is then advanced to the next time an event is scheduled. If there are no events on the event list, the run is terminated.

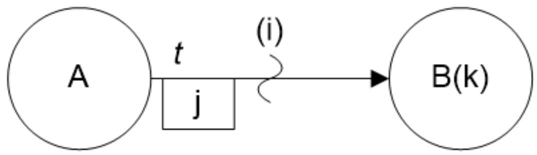
Figure 2. Next-Event Algorithm. Source: Buss (2011).

5. Event Graphs

The information required to describe a DES can be logically represented in an event graph. Schruben first introduced the concept of an event graph for DES in 1983 and suggested that the lack of event graphs prior to this time “perhaps contributed to the perceived sophistication of the event-scheduling approach to discrete event system simulation” (Schruben 1983, 957).

An event graph concisely displays events and their scheduling relationships using nodes and directed edges. The scheduling edges may contain conditions for the scheduling to occur, a time delay, and parameters to pass to the event being scheduled. Each event can show what variables it expects to receive and what state variables will be

modified when that event occurs. All this information is compactly displayed in a scheduling edge with nodes that serves as the fundamental building block for an event graph. Figure 3 depicts a sample scheduling edge.



In this diagram, event "A" schedules event "B" after time delay "t" if condition "i" is met. Event A will pass parameter "j" and event B will receive this parameter and refer to it as "k." More than one parameter may be passed and received, but the order must be consistent between the passing event and receiving event. Additionally, more than one condition may be specified and conditions may be linked with "and" or "or" conditions.

Figure 3. Scheduling Edge with Arguments and Events with Parameters. Source: Buss (2011).

B. SIMKIT OVERVIEW

Simkit was developed by Dr. Arnold Buss at NPS. It was "designed with a pure discrete event world view" (Buss 2002, 243) enabling a straightforward transition from an event graph to implementation in code. "Based on proven Event Graph methodology, Simkit has been used to quickly create models in a wide range of areas, including logistics and operational support, undersea models, and models that evaluate algorithms for allocation of weapons and sensors to targets in ground combat" (Buss 2004). The program is written in the Java programming language that enables use on virtually all modern operating systems and is copyrighted under the GNU public license which allows for open source and free distribution (Buss 2002). Simkit has successfully supported numerous theses and research efforts at NPS and continues to evolve to meet an expanding variety of problem sets.

Simkit is a logical tool for building RSIM for a number of reasons. Most commercial simulation products require either a seat or site license fee to employ the software. Additionally, many products have a fee structure that incurs additional costs to run simulations with a high number of entities—which RSIM certainly has. By using Simkit, the licensing fee is removed and there are no concerns about RSIM being

rendered useless for future studies due to funding. Because the supply domain fits nicely into the DES realm that can be naturally described with an event graph, Simkit's natural linking to event graph methodology provides a natural transition to implementation and continued modifications as additional features are added. Finally, the fact that the simulation is implemented directly at the Application Programmer Interface instead of a graphical user interface allows additional flexibility for the programmer to incorporate additional features as desired. It is important to note that the automated output and post processing in Simkit is limited to very basic statistical measures. To develop the analysis detailed in Chapter IV, we outputted data in a form easily ingested by other programs to conduct in-depth analysis after the simulation was complete.

C. RSIM INTRODUCTION AND SCOPE

RSIM was developed as a DES in Simkit to help verify NAVARM outputs and provide additional insights for decision makers and analysts. RSIM simulates failures at the individual part level and then aggregates up to the individual WS level to help assess the accuracy of EBO and WS availability in NAVARM. To simulate the system of interest, three major classes of entities are created: parts, WS and part positions. Each part has attributes that include:

- Status (i.e., functioning or down for maintenance/supply),
- Planned failure time (detailed below), and
- Position (specifying where it is installed if currently in use).

Each WS has attributes that include:

- Type (e.g., CH-53 helicopter),
- Availability status (i.e., up or down), and
- A list of part positions that comprise the WS (e.g., utility hydraulic pump).

A part position has attributes that include:

- The WS (if currently in use),
- Parameters describing expected failure times, and

- Parameters describing the time for a working part to return to inventory after breaking.

Modeling failures in a manner that closely mirrors reality is crucial to attaining realistic outputs. Expected failure rates can be derived from existing databases and are broken down into failures that can be repaired at the site and failures that cannot. Some parts have only one type of failure or the other while some have both.

RSIM tracks the type of failure to later develop an expected time the part will return to inventory in a working status. To handle the difference in types of failures, RSIM first adds the failure rates then assigns a failure time based on the combined rate. When the failure occurs, a random number draw is compared to the ratio of repairable and non-repairable failures to assign the type.

While multi-parameter distributions such as the Weibull that allow specification of mean and variance are generally preferred for detailed modeling of failure rates, the databases used for RBS currently provides only the mean failure rates. As a result, RSIM employs the exponential distribution to generate a stochastic failure rate. This is also consistent with the assumptions established for both NAVARM and SPO.

Failure rates in the database and RSIM are specific to a part position on a WS type; for example, a hydraulic pump used on a CH-53 utility system may have a different failure rate than the same type of hydraulic pump used on an SH-60S utility system. In fact, the same pump may be installed on different WS types between failures and thus have different failure rates assigned based on where it is installed. When a part fails, RSIM immediately orders a new part and then removes the part from the usable pool for a specified period of time until it is repaired or resupplied. This is an implementation of the (S-1, S) inventory policy discussed above. The expected times for repair or resupply are given in the available databases and are currently modeled deterministically in RSIM.

While RSIM’s core logic is best described with an event graph, the basic steps can be summarized as follows:

- Read data in from database and instantiate all entities specified in the data.

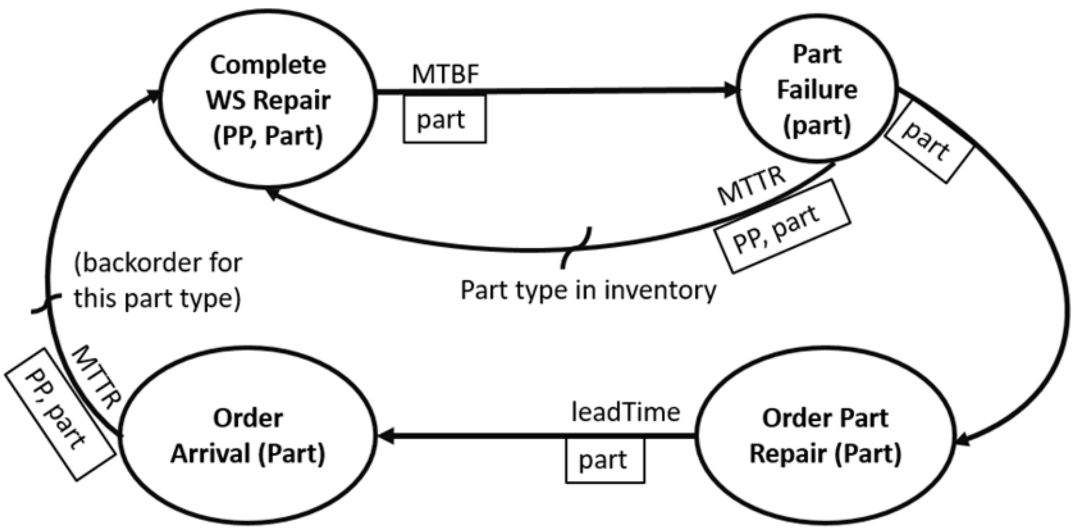
- Assign parts to fill each WS and assign a first failure time stochastically for each part based on the specified distribution.
- When a failure occurs:
  - Assign a time the part will return to service.
  - If a part of the correct type is available in inventory, decrement the inventory and then place WS in down status for the specified MTTR. If a part is not available, add the WS to a first in first out (FIFO) queue for that part type.
- When a part returns to a ready-for-issue status at the site, use it to repair the first WS in the FIFO queue awaiting that part type. If no WS are awaiting that part type, return the part to inventory.

To manage complexity, simplify the verification and validation process, and ensure acceptable simulation run-time, RSIM tightly scopes the factors considered in the simulation. RSIM also maintains flexibility to add new factors as desired to more closely mirror reality or support future study objectives.

In its current state, RSIM ingests summary level data on flight hours, failure rates, repair times and shipping times, and most of these factors are treated deterministically. While RSIM could simulate actual flight sorties and assign failures based on WS flight times, the effect of this added fidelity would likely be nominal when considering inventory policies and thus is not included. Likewise, scheduling the repair process at intermediate and depot level and including manpower and part availability consideration here would also have minimal effect on the metrics currently of interest; expected values are used in lieu of this detailed analysis.

D. RSIM EVENT GRAPH AND IMPLEMENTATION

Figure 4 depicts a simplified event graph describing the overall model of part failures and subsequent repairs in RSIM. This version of the event graph is intended to provide a broad understanding of the flow of parts through the system. A more detailed description of activities and state variable updates occurring at each event is provided in this section.



RSIM simulates real world part failures and the ensuing repair process through a series of events. This figure depicts a simplified event graph to show the broad flow of events over time. The circles represent events and the arrows represent scheduling edges. In some instances, the scheduling edges have a Boolean condition that must be satisfied for the scheduling edge to schedule the next event. The scheduling edges may also have delay times and parameters to pass. A detailed description of each event is provided in the text.

Figure 4. Simplified RSIM Event Graph

The RSIM event graph depicted in Figure 4 is implemented in the Java programming language using the Simkit library (Buss 2002, 2004). Simkit provides the necessary support for converting the event graph into working code. An additional open source library, UCanAccess (2017), is used to interact with the MS Access database inputs provided by NAVSUP.

Every Simkit simulation begins with a *Run* event (not depicted in Figure 4). This event occurs at simulation time zero and serves to initialize all state variables in the system. Because a simulation can be run for numerous replications sequentially (without user input) the *Run* event must contain all code required to achieve starting conditions from any state. RSIM uses this event to place a part into each part position on each WS so that all WS are "up" (i.e., full mission-capable) status at the beginning of the simulation. The process of achieving a steady state before collecting metrics is described in Chapter IV. The *Run* event schedules *Part Failure* events for each installed part at a specified time based on available data for the part position. To calculate the next failure

time, RSIM calculates an expected failure time for the given part position and provides this as a parameter for a random draw in the corresponding exponential distribution. A given part position will have the same MTBF for each WS of that type. If random draws were conducted from the corresponding exponential distribution for each of these WS, we would have several scheduled failures clustered around the mean. Only after several failures had occurred at each of these part positions would the expected failure times be distributed like we would expect in steady state. In the case of a very long MTBF, it would take a long time for the spread of failure times for that part position to reach steady state. To reduce the warm-up time, the initial failure times are distributed throughout the time span of the expected MTBF. For example, if there are 20 WS of type A and part position B has an MTBF of 100, the part position in the first WS of type A would be scheduled by drawing from an exponential distribution with mean five; the second WS would use a draw from exponential distribution with mean ten; etc. At the completion of the reset and run event, the event list has one failure scheduled in the future for every part installed on a WS.

The *Part Failure* event receives the specific part that failed as a parameter and performs all activities required to simulate this failure. First, the part status and associated WS status are both set to “down” (i.e., non-mission capable). The *Failure* event schedules an *Order Part* event to occur immediately and passes the part as a parameter. If a part of the same type is available in inventory, the *Failure* event schedules a *Complete WS Repair* event to occur in the future by the MTTR time units associated with that WS and the *Failure* event passes the new part and the part position of the failure as parameters. If a part is not available in inventory, the *Failure* event adds the part position to a FIFO queue for the associated part type.

The *Order Part Repair* event simulates a simplified view of acquiring parts from the supply site point of view. Because this is an (S-1, S) policy for RBS parts, the supply system will immediately turn the part in to the system and receive a ready-for-issue part when it becomes available either through repair or resupply. This event calculates an expected lead time and schedules the *Order Arrival* event for this part. To determine the lead time, a random number is drawn and compared to the ratio of repairable parts to

determine whether the lead time should be calculated for an onsite part repair or a resupply from the depot.

The *Order Arrival* event simulates the site supply system receiving a ready-for-issue part. When the part is received, the inventory for the corresponding part type is incremented. If there is a backorder for this part type, the first part position in the FIFO queue and the part are sent as parameters to the *Complete WS Repair* event which is scheduled with an MTTR delay.

The *Complete WS Repair* event simulates installation of the given part in the given part position. This event generates a new failure time for the part using a random draw from the exponential distribution with a mean based on the part position. Finally, the *Complete WS Repair* event checks all part positions on the corresponding WS to see if they have parts assigned; if all part positions have associated parts, the WS status is marked as "up."

E. RSIM ASSUMPTIONS

We make a number of assumptions in the RSIM implementation, some of which could significantly impact the results. These assumptions are made for a variety of reasons to include limited data availability, code simplicity, and reduced run-time. The inherent flexibility in RSIM implementation makes these assumptions fairly easy to modify or eliminate through code manipulation. The following are significant assumptions currently made in RSIM:

- Failure rates are accurately represented by an exponential distribution—as stated early, failure rates would likely be better represented with a Gamma or Weibull distribution, but the limited failure data provided does not allow implementation of a multi-parameter distribution. The exponential distribution is not well suited to represent wear out failures that occur at fairly predictable intervals as opposed to "memoryless" failures.

- Failures are independent—because failure times are scheduled into the future on a continuous timeline and there are no dependencies programmed in, simultaneous failures will not occur despite real-world experiences that suggest otherwise.

- Failures in the simulation should continue to happen when the WS is down—while failure rates in the database are given per flight hour, this

data along with average flight hours is used to develop expected mean time between failures. Although parts are much less likely to fail when the WS is out of service, scheduled failures continue to occur in the simulation to ensure the expected failure rate is maintained. This may result in overlap of delay times for backordered parts. With a higher fidelity data set, this could be improved by developing conditional probabilities that better reflect the empirical data.

- Expected sub-indentured part failure times are not reset when a parent part is changed—this assumes all parts are repaired and that when they undergo repair, it does not affect reliability of the separate sub components. This assumption will fail if the part is resupplied and sub components are not salvaged, but the available data does not delineate how often parts are repaired when they go off-site and what happens to sub-indentured parts when a resupply is necessary for the parent part. Of note, this assumption will lead to a conservative estimate of availability, though the extent of the impact is unknown with the data currently available.

- Demands are FIFO—this assumes that no priority will be given to WS of types that are below their availability goal or some other prioritization scheme.

- No lateral resupply—there is no cross-leveling between sites that have high inventory and sites that have low inventory or backorders for a particular part.

- Cannibalization is not allowed—while cannibalization (removing parts from a down WS to another WS to return it to an up status) is practiced in the real-world, NAVARM is designed to achieve desired readiness states without using this extreme measure and thus RSIM does not allow it either.

- Repair times are independent—RSIM does not attempt to simulate backlogs in the repair pipeline that would likely occur if multiple parts of the same type were in the repair pipeline simultaneously.

F. RSIM DATA

RSIM is designed to ingest data in currently available data structures provided by NAVSUP. RSIM is configured to read inventory levels from the provided database or from a separately provided CSV file. RSIM uses UCanAccess as a third party open-source driver to input the NAVSUP supplied Microsoft Access database. This program runs in Java and executes standard Structure Query Language queries on the database to

retrieve the data of interest for RSIM. While the supplied databases contain data on both RIMAIR and RBS parts, optimization tool output is used to set RBS part allowances. As such, the calculations and RSIM analysis will ignore the RIMAIR WS at this time.

1. Input Data

The two values in RSIM driven by NAVSUP supplied data are the part failure times and the lead time associated with a part returning in a ready-for-issue status.

The calculation for part failures requires the following values from the NAVSUP supplied database:

- MRF—This value from the "Candidate" table represents specific part position failures per maintenance cycle that cannot be repaired on site
- RPF—This value from the "Candidate" table represents specific part position failures per maintenance cycle that can be repaired at the site.
- War_FHRS—This value from the "ParamSW" table represents the total maintenance cycles per quarter for each WS type at the site.
- WS_NUMBER—This value from the "ParamSW" table represents the total number of WS of the corresponding type for each WS type at the site.
- Quantity per application (QPA)—This value from the "Candidate" table represents the quantity per application for a given part position. This serves as a way to condense the database and represents multiple parts of the same type and indenture level with a single line in the database (e.g., 100 rivets on a radio may be represented with a single part position and a QPA of 100).

With these values from the database, the calculations below transform the data into expected failure time (in days) for each part position. First, we calculate the expected flight hours per day as follows:

$$WS\_FltHours = \frac{100}{90}\frac{War\_FHRS}{WS\_NUMBER}. \tag{3.1}$$

Of note, we use the 100/90 factor to translate the units from maintenance cycles (100 flight hours) per quarter to flight hours per day. Next, we calculate the mean time between failures in hours as follows:

$$MTBF = \frac{100}{(RPF + MRF)QPA}. \qquad (3.2)$$

We use the 100 in the numerator of the MTBF formula to translate from maintenance cycles to hours. Finally, we can calculate the expected failure time in days (due to repairable or non-repairable failures) as:

$$ExpFailure = \frac{MTBF}{WS\_FltHours}. \qquad (3.3)$$

The next data objective is to calculate lead times for parts that are repairable on site and parts that are not repairable on site. The calculations for lead times require the following values from the database:

- IMA_RPR_TM—This value is in the “Candidate” table and represents the total time to repair a part on site for a given part position. No shipping time is required for this.
- HP_OST—This value is in the “Candidate” table and represents the order and ship time required for parts not repairable on site for a given part position.
- WHSL_DELAY—This value is in the “Candidate” table and represents the average time to repair or acquire a new part for a given part position when it cannot be repaired on site.
- MTTR—This value is in the “ParamWS” table and represents the time required by the unit to swap a bad part for a good part on the WS for each WS type. While not used here for the lead time calculations, this delay is used as described above in scheduling a delay time for WS repair.

With these values from the database, the calculation for lead times is fairly straightforward. If the part is repairable on site the lead time in days is:

$$LeadTime = IMA\_REPAIR\_TM\ . \qquad (3.4)$$

If the part is not repairable on site, the lead time in days is:

$$LeadTime = HP\_OST + WHSL\_DELAY\ . \qquad (3.5)$$

2. Output Data

RSIM is a flexible tool capable of creating a large variety of output metrics. To reduce run time and code complexity, only metrics germane to readiness are calculated at this time. In most cases, adding additional metrics is fairly straightforward allowing the tool to expand in scope to answer additional questions that may arise. Currently, RSIM provides the following outputs:

- Mean backorders by part type,
- Mean inventory on-hand by part type,
- Fill rate by part type,
- Average A_o by WS type (i.e., the average number of WS by type with all parts functioning), and
- Percent of time A_o at or above availability goal by WS type.

IV. OUTPUT ANALYSIS

RSIM outputs several metrics by WS and part type to allow comparison to other RBS optimization software (i.e., NAVARM, SPO, and ARROWS). Additionally, it provides decision makers with a more comprehensive understanding of what to expect if a set of recommended inventory levels is used. For each WS type, RSIM provides the mean number of WS available, the corresponding readiness rate, and the percent of simulated time the WS type was at or above its given readiness goal. For each part type, RSIM outputs the mean on-hand inventory level, mean number of backorders, and the fill rate.

The primary metric of interest is the readiness rate by WS type. Given the crucial nature of having required force levels available at any given time, NAVSUP must ensure the inventory quantities selected will enable this objective. RSIM, NAVARM, SPO, and ARROWS each have assumptions built in that may not be accurate in every situation, but a comparison of the outputs can be helpful in assessing the validity of the readiness estimates.

The data used for the analysis in this chapter was generated using a Dell Inspiron I5378 laptop running Windows 10 with an Intel Core i7-7500U 2.7 GHz CPU and 8 GB of RAM. RSIM is implemented in JDK 1.8 and utilizes 64-bit Simkit version 1.4.6 and UCanAccess version 4.0.1. Run times for RSIM with 10,000 simulated days and 30 replications range between 2.5 and 59 minutes for the seven sites analyzed. NAVARM runs were conducted on the same laptop described above using the 32-bit version of Microsoft Excel 2016. There are several NAVARM settings that affected run time. For this analysis, we set NAVARM to complete 10 main passes to modify the ordering and 10 polishing passes (described in Chapter II) for each run resulting in run times between 30 seconds and 18 minutes.

A. ANALYSIS PARAMETERS

Although RSIM uses random variable inputs with known distributions calculated based on historical demand, the distributions of the output variables are unknown. RSIM

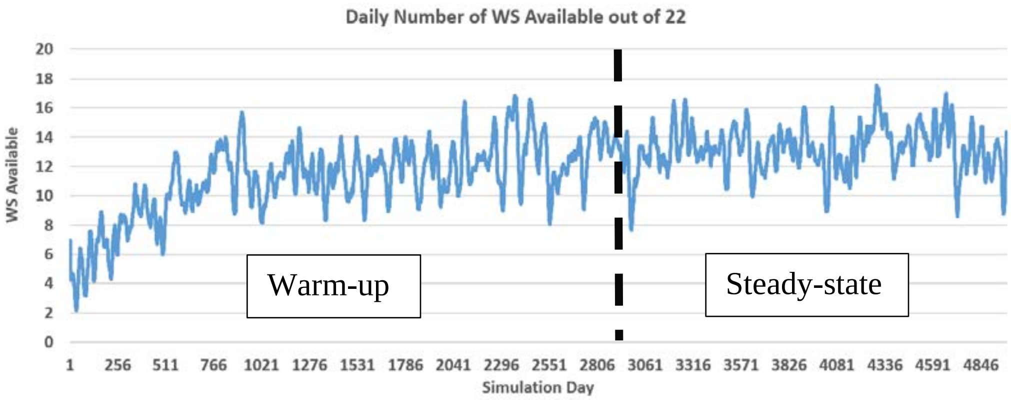
applies standard statistical methods to estimate percentiles, means and variances for the metrics of interest. Additionally, it calculates confidence intervals as measures of uncertainty for the means.

While the calculations for these metrics are standard and well known, they all have an embedded assumption that the samples are independent and identically distributed. Unfortunately, this assumption of independence will not hold when examining simulation data without some additional measures. In fact, there is a large degree of correlation between neighboring observations in a simulation. For example, if a radio fails on day 10 in the simulation and encounters a longer than usual delay to receive a resupply, a subsequent radio failure of the same type on day 11 is more likely to also encounter an extended delay for repair.

Another issue with simulated values concerns the starting conditions that do not necessarily reflect the system's steady state. By starting the simulation with all stocks set to their suggested stock levels and all WS in a full mission capable status, analysis on data starting at day zero of the simulation will bias the estimates of readiness. Because this is a non-terminating simulation (i.e., operations run continuously), the simple solution to remove this bias in the metric estimates is to allow the simulation to achieve a steady state prior to collecting data for analysis. The challenge is identifying a warm-up period that allows the simulation to achieve steady state without wasting more computing resources than necessary. One technique to identify an appropriate warm-up period is to run numerous replications and sample at several intervals (e.g., the 50th, 100th, 150th … days) and compare the histograms to see where they start taking on the same shape. In the ABC's of Output Analysis, Sanchez recommends the final cutoff for deletion of a warm-up period be an even number to reduce suspicion of data manipulation (Sanchez 1999, 26).

To identify an appropriate truncation point for RSIM data collection, we collected data on the number of WS available at the beginning of each simulated day and located the point where readiness was in a steady-state. Although readiness rates fluctuated significantly over time, there is a point at which there is no longer an upward or downward trend in the readiness rate. Figure 5 depicts daily readiness levels for a

selected WS type at a representative site. We selected a WS type with 22 individual WS and employed a 10-day running average of available WS to make the trend more apparent. Although it is impossible to select a precise time when RSIM achieves steady-state and the trends vary by WS; it appears likely that steady-state is achieved at approximately 1,000 simulated days; however, there may still be a slight upward trend until almost 3,000 simulated days. Similar trends were noted in other WS types and sites. To ensure a bias is not introduced, we conservatively selected 3,000 simulated days for the truncation point in the RSIM runs used for the analysis in this chapter.



To ensure a bias is not introduced in the RSIM output from starting condition levels, a truncation point is selected that will allow the system to reach a steady-state before data collection begins. Here we depict daily readiness levels for a WS type with 22 individual WS and employ a 10-day running average of available WS to make the trend more apparent. This WS arrives at steady-state between 1,000 and 3,000 simulated days. We select a conservative truncation point of 3,000.

Figure 5. Daily Readiness Levels for a WS Type with 22 Individual WS at a Representative Site

After removing the bias induced by samples collected during the warm-up period, we must account for the correlation between samples to ensure confidence intervals are accurate. One way to ensure sample statistics are independent of each other is to utilize replication-deletion. This method involves collecting each sample from an independent replication and removing data collected prior to steady-state. This is the safest and most straightforward method, but also the most computationally intensive due to the requirement for a warm-up period at the beginning of each replication.

There are three alternative methods that allow for one long simulation run and only require data deletion for one warm-up period. The first method is batch means. A batch refers to a period of time over which data is collected. For this method, the analyst identifies a proximity for which the correlation between two selected points in time is nearly zero (the correlation drops as proximity decreases). The batch sizes should then be at least five times this proximity to prevent confidence intervals from being overly optimistic (Welch 1983, 307). These batches can then be treated as independent and we can calculate statistics as if they were independent replications. The second option is the regenerative method. This method identifies a point where all conditions will match exactly several times in the course of a simulation run (i.e., every part and WS has the exact same status) and every time this point is encountered, a new batch of data is collected. This requires a certain probabilistic structure and does not lend itself to the simulation built for this thesis. The final technique is the spectral method which involves dealing with the correlation directly instead of attempting to eliminate it. Generally this is done using an autoregressive time series model fitted to the output or regression techniques applied to the log of the periodogram or sample spectrum (Welch 1983, 320).

While the batch means approach is an acceptable method of dealing with the correlation in RSIM data, we employed the replication-deletion method for the analysis in this thesis because it is conservative and straightforward to implement. The run-time and number of replications required make this an acceptable choice for this study, but the batch means approach may be appropriate for future analysis using RSIM.

Based on steady state analysis conducted for several sites and the desired margin of error, we used a warm-up period of 3,000 simulated days before collecting 7,000 simulated days of data for 30 replications at each site analyzed. These simulation settings achieved readiness estimates with a margin of error under 0.5% for each WS type at all sites.

B. DATA SETS TESTED

RBS is designed to select optimal inventories at the site level. For the purposes of this study, a naval site may consist of a base or a ship with tenant aviation WS. The size of these sites, the number of tenant WS, and the logistical challenges can vary widely from site to site.

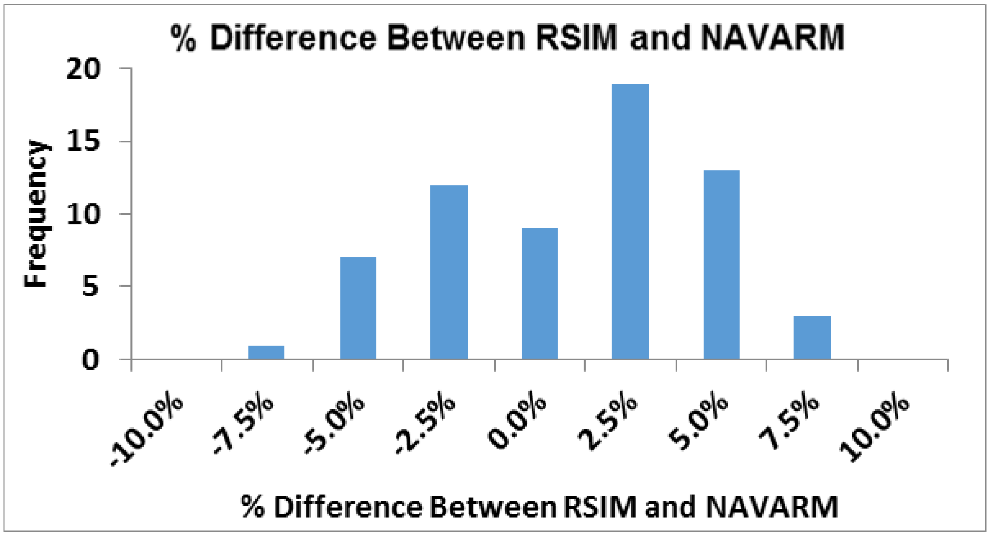
We selected seven representative sites to conduct the analysis for this thesis. Four of these sites are shore-based while the other three are ship-based; of the three ship based sites, two are comprised primarily of Marine Corps WS. Four sites are composed primarily of Navy WS while the other three are primarily Marine Corps WS. The number of WS types at these naval sites ranges from 3 to 23 with a mean of approximately 9 WS types and 111 individual WS per site.

C. COMPARISON OF NAVARM AND RSIM RESULTS

Because NAVARM and RSIM employ different methods to estimate expected readiness levels, a comparison of the output can help verify the results. Since the accuracy of RSIM has not been validated using fleet data, we cannot treat the output as the authoritative solution. Instead, we use the results here to help assess the likelihood that a given output is accurate. If both NAVARM and RSIM arrive at similar solutions using different methodology, we can use that information to help verify the implementation of each model.

1. Readiness Comparison

To compare readiness estimates in NAVARM to the observed readiness in RSIM, we calculated the difference in readiness for the 64 WS types at the seven sites analyzed. Figure 6 shows the summary histogram of the differences in expected readiness for the 64 WS types tested. Out of the 64 WS types analyzed, 53 have expected readiness levels within 5% and the mean difference for all WS types in this sample is 0.2% with no systemic bias to over or under estimate readiness noted.



This figure shows the summary histogram of the differences in expected availability between RSIM and NAVARM for the 64 WS types tested. Out of the 64 WS types analyzed, 53 have expected readiness levels within 5% and the mean difference for all WS types in this sample is .2% with no systemic bias to over or under estimate readiness noted.

Figure 6. Difference between RSIM and NAVARM Readiness Estimates for 64 WS Types at Seven Sites

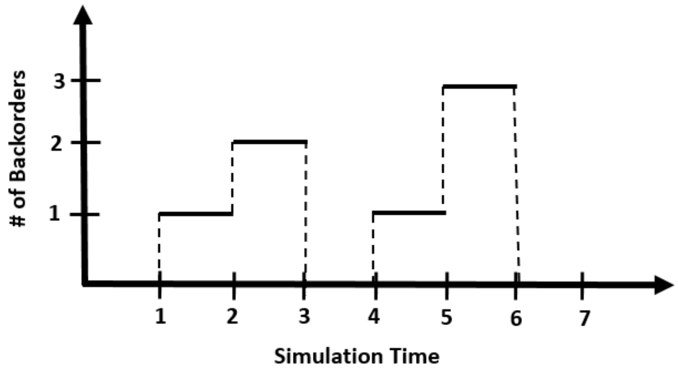
While the difference between expected readiness given by RSIM and NAVARM is likely acceptable in the current versions, we have tried to identify key drivers of any differences found in the hope of further explaining the differences and ideally reducing the errors. First, we consider attributes of the WS type that may complicate calculations for readiness in the models. The factors of interest by WS type are:

- Sum of QPA for the WS type: QPA is described in Chapter III.
- Commonality: This is a measure of how often the same part type is used throughout the site. This adds a layer of complexity in the optimization model as changes in inventory can affect numerous WS types. For the purposes of this analysis, we tally the number of times the same part type is found in other part positions and then sum across all part positions for each WS type.
- Number of parts: The number of part positions tracked on a given WS type in our data sets ranged from 80 to over 8,000.
- Indenture depth level: While NAVARM uses the negative binomial distribution to model the number of failures at intermediate indenture levels, RSIM assigns failures at the individual part level and tracks their impact on the metrics of interest.

We looked for correlation between each of the factors listed above and the difference in readiness estimates produced by NAVARM and RSIM. None of the correlations were significantly strong, with total number of parts being the highest at 0.49, indenture depth level and mean number of common parts slightly lower at 0.43 and 0.40, respectively, and QPA being clearly non-significant at 0.02. The total number of parts and average indenture depth level for a WS type are strongly correlated at 0.78 making it difficult to assess whether one or both of these factors are a driver in the difference in readiness.

2. EBO Comparison

EBO is a metric for the average number of backorders expected in the system for a given part position or part type at a given site. This metric depends on the failure rates and inventory levels as well as delay times to receive a new part after an order is placed. This metric takes into account both the volume of backorders and the time a backorder remains unfilled. Figure 7 is a small graphical example of backorders for a particular part type. The dotted lines represent times when the level of backorders change due to a new backorder placed or an order arrival while the solid line represents the length of time the number of backorders remains at a given level. To calculate the EBO, we simply calculate the area under the curve and divide by the total time. RSIM sums the area under the curve for each part type and part position through the course of the simulation and outputs all EBO. By contrast, NAVARM uses the VM model to calculate the pipeline for each part and uses this to attribute EBO levels to each part position that are ultimately rolled up the indenture tree to the WRA level.



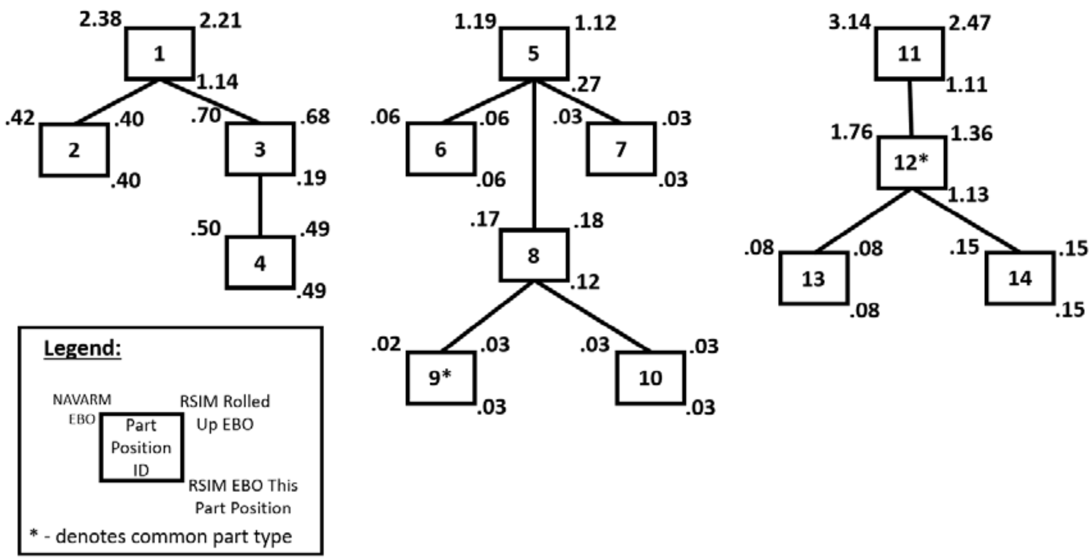
This simplified image demonstrates the concept of a time varying statistic. When we calculate EBO, we are interested in both the volume of backorders and how long they go unfilled to calculate the average backorder level. To calculate the EBO for this small example, we sum the area under the curve (7) and divide by the total simulated time (7) and return an EBO of 1.

Figure 7. EBO Calculation Example

Because EBO play an integral part in the NAVARM calculations of expected readiness, we configured NAVARM and RSIM to output EBO for every part position to compare expected EBO levels. The Expected Pipeline NAVARM calculates for each part position includes the raw pipeline contribution from that part plus the EBO contribution of its children. The EBO levels NAVARM outputs for each part position is calculated based on the expected pipeline which determines the distribution of backorders. By contrast, RSIM outputs the EBO contribution of each part position individually and only rolls up the EBO levels by part type. To compare NAVARM and RSIM EBO levels, we simply summed the EBO levels up the indenture tree to allow comparison of every part position.

To demonstrate the EBO comparison methodology, we ran a small toy problem and present the results in Figure 8. This toy problem represents a notional WS with three WRAs, three levels of indenture, and one common part. Each box represents a part position on a WS. The number on the top left is the EBO level output by NAVARM for that part position and represent the EBO for that part position and any contribution from its children. The number at the bottom right is the EBO level output by RSIM and represents the average observed EBO level for that part position through the course of the simulation. The number at the top right represents the rolled up EBO level in RSIM for

comparison to the NAVARM value. This number is simply a sum of each child's rolled up EBO level plus the EBO contribution of the given part position.

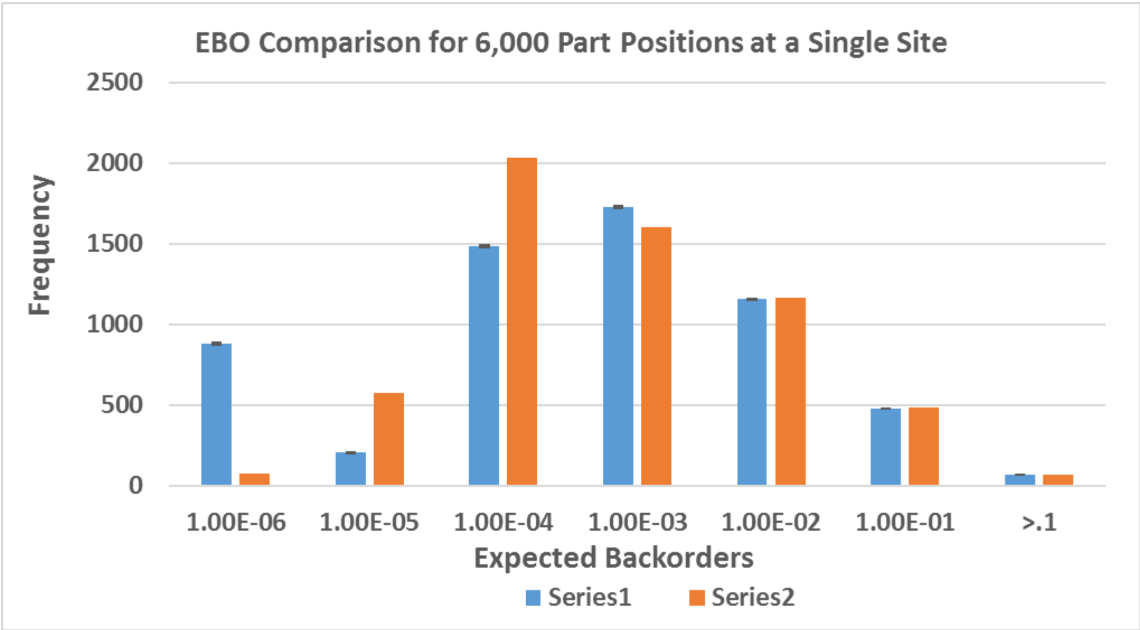


To demonstrate the EBO comparison methodology, we ran a small toy problem and present the results in Figure 9. This toy problem represents a notional WS with three WRAs and three levels of indenture with one common part. Each box represents a part position on a WS. The number on the top left is the EBO level output by NAVARM for that part position and represent the EBO for that part position and any contribution from its children. The number at the bottom right is the EBO level output by RSIM and represents the average observed EBO level for that part position through the course of the simulation. The number at the top right represents the rolled up EBO level in RSIM for comparison to the NAVARM value. This number is simply a sum of each child's rolled up EBO level plus the EBO contribution of the given part position.

Figure 8. EBO Comparison between NAVARM and RSIM on a Small Toy Problem

We used this methodology to examine the differences between NAVARM and RSIM EBO levels for an actual site. Figure 8 is a histogram of EBO levels for 6,000 part positions at a representative site with seven different WS types. This site has over 11,000 part positions tracked; of these, fewer than 8,500 have EBO levels greater than zero in RSIM or NAVARM. Figure 9 charts the 6,000 part positions with the highest EBO levels. While there are some differences noted between RSIM and NAVARM levels in this histogram, parts with extremely low EBO levels will not significantly impact overall WS readiness. Here we note that the counts are nearly identical for EBO greater than 0.001. Further, the magnitude of the difference is generally negligible with only a 3.4%

difference in the sum of EBO for NAVARM and RSIM and an average difference of 0.0003 per part position.



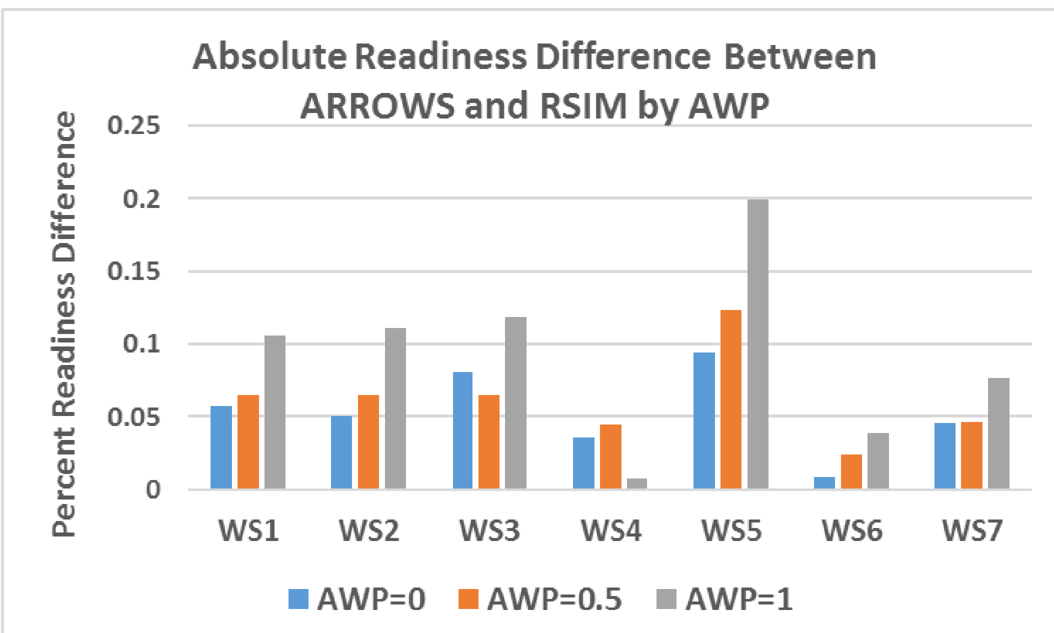
This histogram provides a summary of EBO levels for RSIM and NAVARM at a representative site with seven WS types. While we note some differences for the EBO at the lower end, we are primarily interested in part positions with larger EBO which will have more impact on the WS readiness. Here we note that counts are a very close match for EBO levels >0.001. Overall for this data set, the sum of EBO for NAVARM and RSIM are only 3.4% different and the average difference per part position is 0.0003 for this site.

Figure 9. EBO Comparison between RSIM and NAVARM at a Single Site

While the difference in EBO levels is relatively small and does not significantly impact readiness estimates in NAVARM, we examine the correlation of part position attributes with the difference in EBO levels. In particular, we are interested in whether the indenture level of a part position is correlated with the EBO difference. A high correlation here would suggest the VM assumption of negative binomial distribution for modeling EBO of sub-indentured parts is invalid. Instead, we note a low correlation of 0.09 suggesting the negative binomial assumption employed in NAVARM is acceptable.

D. COMPARISON OF AVAILABLE RBS TOOLS

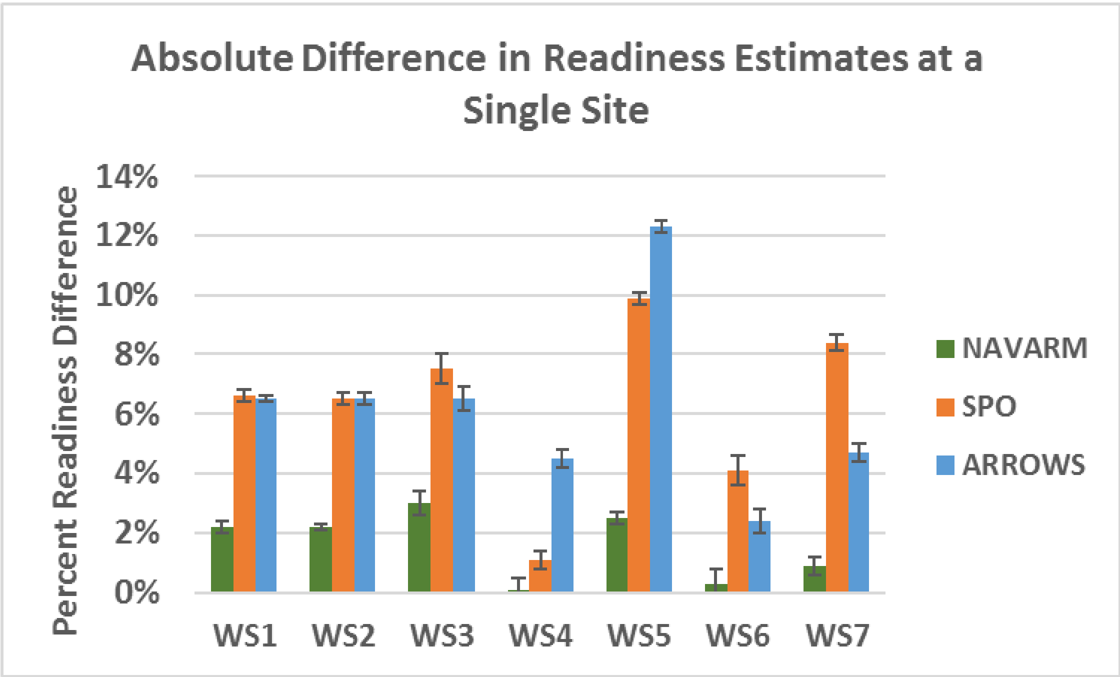
As NAVSUP considers whether to switch RBS optimization tools, it is crucial for the decision makers to assess the accuracy of the NAVARM expected readiness calculations and compare them to the SPO and ARROWS tools currently in use. Because RSIM models the system at the part and WS level, its method of observing readiness rates through the course of a simulation provides an independent observation to compare against the optimization tool estimates available. SPO, ARROWS, and NAVARM were run at a representative site with seven different WS types and a total of 62 WS. ARROWS employs an Awaiting Parts (AWP) weighting factor that accounts for overlap of SRA backorders on WRA downtime; this factor is set between 0 and 1. NAVSUP uses a default value of 0.5 for AWP. Figure 10 shows the difference between RSIM observed readiness for the given site and the ARROWS estimated readiness at three different AWP settings. Here we note that the effect of changing AWP from 0 to 0.5 is negligible. For the remainder of our tool comparisons, we use an AWP setting of 0.5 for ARROWS.



This chart depicts the difference between readiness observed in RSIM and the estimate produced by ARROWS with different AWP settings for a representative site. We note that the difference between AWP=0 and AWP=0.5 is negligible.

Figure 10. Absolute Readiness Difference between ARROWS and RSIM by AWP

We simulated the recommended inventory policies for ARROWS, SPO and NAVARM in RSIM to compare the expected readiness rates for each WS type at the site described above. Figure 11 shows a summary of the resulting differences in estimates. In this case, it becomes clear that NAVARM's estimated readiness rates are much closer to RSIM than SPO or ARROWS estimates are.



We simulated the recommended inventory policies for ARROWS, SPO and NAVARM in RSIM to compare the expected readiness rates for each WS type at the given site. This chart shows a summary of the resulting differences in estimates along with 95% confidence intervals. In this case, it becomes clear that NAVARM's estimated readiness rates are much closer to RSIM than SPO or ARROWS estimates are.

Figure 11. Comparison of Three Readiness Estimates from Available RBS Tools

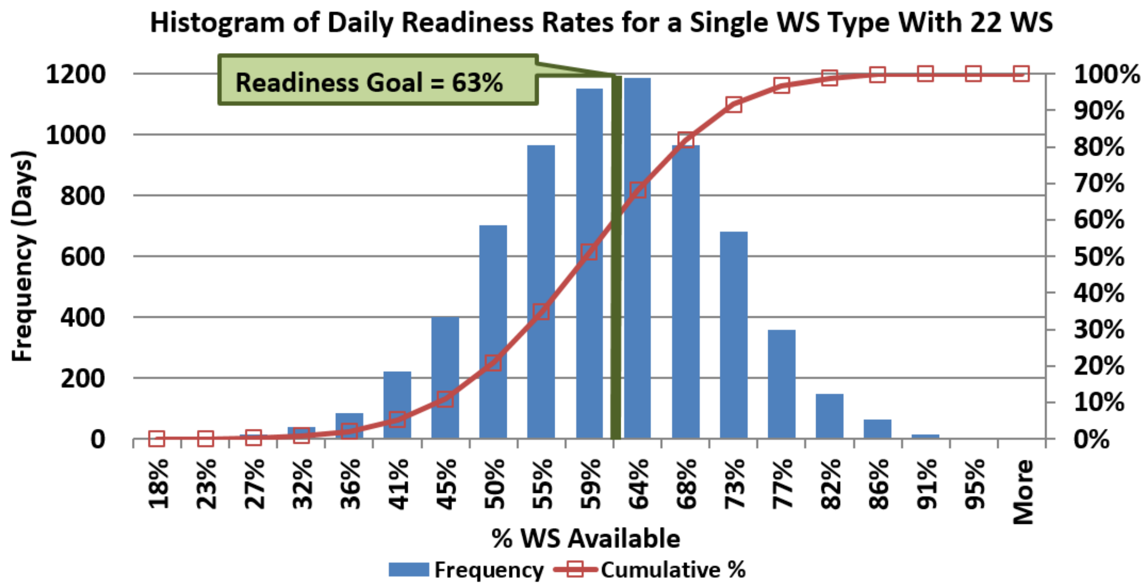
E. ADDITIONAL RSIM INSIGHTS

In addition to verifying NAVARM outputs and providing an independent comparison of the three available RBS optimization tools, RSIM can provide additional insights not available with the optimization output. Here we provide three sample applications where RSIM provides additional insights that may be useful to decision makers: examining daily readiness levels over time, providing metrics by part type, and examining the impact of including RIMAIR WS.

1. Readiness Levels over Time

While the optimization tools only provide the expected readiness levels overall, decision makers may be just as interested in how often the overall readiness falls below other thresholds linked to contingency plans in their area of operations. RSIM provides metrics that include percentage of time above the stated readiness goal and the readiness levels observed at the beginning of each simulated day.

For example, Figure 12 shows a histogram of observed daily readiness levels over a period of 7,000 simulated days for a single WS type with 22 WS at a single site. Even though the most important output of RSIM is the expected readiness achieved (in this case 60.7%, slightly below the 63% goal), additional valuable information can be gleaned: In this simulation run, 48.2% of the simulated time had readiness rates above the goal. A decision maker may be more interested in worst-case scenarios to ensure that assumptions made for contingency planning are realistic. The fact that we expect less than 50% readiness during 11% of the time may be of interest. It is also straightforward to group the readiness rates to identify how often the overall capability in a given category (e.g., strike, lift) falls below a set threshold.



This histogram depicts observed daily readiness levels over a period of 7,000 simulated days for a single WS type with 22 WS at a single site. Even though the most important output of RSIM is the expected readiness achieved (in this case 60.7%, slightly below the 63% goal), additional valuable information can be gleaned: In this simulation run, 48.2% of the simulated time had readiness rates above the goal. A decision maker may be more interested in worst-case scenarios to ensure that assumptions made for contingency planning are realistic. The fact that we expect less than 50% readiness during 11% of the time may be of interest.

Figure 12. Histogram of Daily Readiness Rates for a Single WS Type

2. Metrics by Part Type

RSIM is capable of aggregating observed data in numerous ways to support various decisions and applications. Aggregating by part type can provide analysts with additional insights when making manual adjustments to an optimization output or could provide input for an automated refinement conducted iteratively by the simulation in a future version. Table 1 shows a small sample of output from RSIM for six part types at a single site. The mean backorder level, mean inventory level on hand, number of failures over the period of the simulation and the fill rate are output for each part type. The full output for a single site contains thousands of entries, but an analyst could sort this list to help identify areas where inventory levels could be manually adjusted to incorporate other factors not accounted for in the NAVARM optimization. This process could be automated and take advantage of both NAVARM and RSIM to evaluate the changes.

Moreover, RSIM could be extended to implement its own adjustments and become a complement to NAVARM's optimization.

Table 1. Sample RSIM Output by Part Type

part type	mean backorders	mean inventory	# failures	fill rate
Part 1	0.01	0.89	11	0.91
Part 2	0.37	1.23	1592	0.62
Part 3	0	0.93	7	0.86
Part 4	0.11	0.55	65	0.63
Part 5	0	4.96	6	1
Part 6	0	0.93	6	1

3. RIMAIR Effect on Readiness

Each site has both RIMAIR and RBS parts. As described in Chapter II, stock levels for RIMAIR parts are determined using an 85% Poisson protection level instead of utilizing the RBS optimization methodology. This distinction is used to ensure the RIMAIR parts are adequately resourced. Some part types are both RBS and RIMAIR. The RBS tools (i.e., SPO, ARROWS, and NAVARM) optimize the RBS stock levels separately from the RIMAIR allocations. Unless business rules are established to separate parts for RBS and RIMAIR, we intuitively expect that part commonality will result in RIMAIR allocations affecting RBS part EBO in some cases.

To assess the magnitude of impact on readiness when RIMAIR is included, we re-ran each site in RSIM with the RIMAIR part allocations and WS (with their corresponding part failures) included. We then compared the observed readiness levels with and without RIMAIR included.

Including RIMAIR in the solution had little or no effect in most cases and did not systemically bias observed readiness up or down. Out of the 64 WS observed at the 7 sites, 39 observed readiness levels fell within the 95% confidence interval of the run

without RIMAIR included. The largest difference noted was 5.6% and on average the absolute difference was only 0.7%. Based on this analysis, we conclude that excluding RIMAIR WS from consideration during RBS optimization is unlikely to have a significant impact on the outcome.

V. CONCLUSIONS AND RECOMMENDATIONS

RSIM leverages the strengths of discrete event simulation to develop a comprehensive set of metrics independent of the optimization tools available to NAVSUP. We used RSIM as a method to compare the tools and provide insights to the decision maker. As our understanding of the problem continues to develop, we expect to modify the RSIM assumptions and metrics accordingly to support future needs.

A. CONCLUSIONS

RSIM is designed to help assess the validity of NAVARM outputs, compare NAVARM to other RBS optimization tools available, and provide additional metrics and information that may help decision makers.

Based on the analysis provided in Chapter IV, we conclude the following:

- NAVARM calculations for expected readiness by WS are consistent with observed readiness levels in RSIM and do not systemically over or under estimate readiness.
- NAVARM calculations for expected readiness by WS are more closely aligned to RSIM observed readiness levels than the two RBS optimization tools currently employed by NAVSUP.
- NAVARM assumptions with respect to RIMAIR exclusion and negative binomial employment for EBO calculations appear reasonable for this domain.
- RSIM can be tailored to provide additional insights not available in the RBS optimization tools.

B. RECOMMENDED FUTURE WORK

While RSIM in its current configuration provides utility to decision makers, there are several areas where it can be improved. The following is a list of recommended future work in this realm:

- RSIM currently operates on databases used by ARROWS, SPO, and NAVARM for RBS optimization at the site level. The values in this database (e.g., failure rates, ship times, etc.) are often basic approximations of the actual value. While this database provides a good

starting point for RSIM, an in-depth analysis of maintenance records and other data sources could yield more accurate data and identify appropriate distributions for stochastic modeling of more aspects of the maintenance process.

- Comparing actual readiness rates at several sites to the readiness rates observed in RSIM would help validate the RSIM output. While the near matches in readiness expectations between RSIM and the three available RBS optimization tools builds confidence in the simulation, a comparison to real-world data is necessary to fully trust the RSIM output. Shore based sites may change recommended inventory levels quarterly making it difficult to identify a steady state readiness level for comparison. Ship based sites maintain a more stable inventory policy for longer periods of time making it more feasible to compare RSIM results. This comparison could help determine if the RSIM assumptions are reasonable.

- Varying the data input with a design of experiments could help identify the database elements that most affect observed readiness levels in RSIM. This process would help identify the best variables to focus on in any future data analysis.

- RSIM could be modified to search for additional solutions that may prove better than the inventory levels recommended by the RBS optimization tools. A greedy heuristic, genetic algorithm or other optimization technique could be used iteratively with RSIM to test candidate solutions (proposed by the planner or generated by a heuristic method).

- Each of the assumptions listed in Chapter III could be revisited to conduct sensitivity analysis and/or modify the code to more accurately reflect real world maintenance and supply practices. In particular, RSIM could be modified to reflect traditional cannibalization practices, prioritized queues for resupply, and conditional failure rates by part position.

LIST OF REFERENCES

Buss, Arnold. 2002. “Component Based Simulation Modeling with Simkit.” In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 244–249. San Diego, California: IEEE.

———. 2004. “Simkit Analysis Workbench for Rapid Construction of Modeling and Simulation Components.” Paper presented at the 2004 Fall Simulation Interoperability Workshop, Arlington, Virginia, April 18–23.

———. 2011. “Discrete Event Simulation Modeling.” Unpublished manuscript. https://eos.nps.edu/Simkit/docs/Discrete%20Event%20Simulation%20Modeling.docx.

Buss, Arnold, and Ahmed Al Rowaei. 2010. “A Comparison of the Accuracy of Discrete Event and Discrete Time.” In *Proceedings of the 2010 Winter Simulation Conference*, edited by Björn Johansson, Sanjay Jain, and Jairo Montoya-Torres, 1468–1477.

Cardillo, John. 2016. “Readiness-Based Paring.” Presentation to Javier Salmeron, Mechanicsburg, Pennsylvania, 27 April 2016.

Chief of Naval Operations. 2003. *Operational Availability of Equipment and Weapons Systems*. OPNAV Instruction 3000.12A. Washington, DC: Department of the Navy: Chief of Naval Operations.

———. 2011. *Readiness Based Sparing*. OPNAV Instruction 4442.5A. Washington, DC: Department of the Navy: Chief of Naval Operations.

Logistics Management Institute. 2008. *Defense Sustainment Chain Operational Readiness Evaluator: User Manual*. McLean, VA: Logistics Management Institute.

Moulder, Roger D., T. J. O’Malley, and David K. Peterson. 2011. “Sizing Inventory with Readiness-Based Sparing.” *Air Force Journal of Logistics* 35: 18–29.

Salmeron, Javier. 2016. “Readiness-Based Sparing Inventory Optimization Model.” Presentation to Navy Supply Systems Command, Monterey, California, 6 October 2016.

Sanchez, Susan M. 1999. "ABC's of output analysis." In *Proceedings of the 31st conference on Winter simulation: Simulation*, edited by P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans. Volume 1, pp. 24-32. New York: Association for Computing Machinery.

Schruben, Lee. 1983. “Simulation Modeling with Event Graphs.” *Communications of the ACM* 26(11): 957–963.

Sherbrooke, Craig C. 2004. *Optimal Inventory Modeling of Systems: Multi-echelon Techniques Second Edition*. New York: Kluwer Academic Publishers.

Strauch, F. C. 1986. *ARROWs Model Evaluation*. DTIC Report No. AD-A173869. Mechanicsburg, PA: Operation Research Department, Navy Fleet Material Support Office.

UCanAccess. 2017. Accessed April 15, 2017. https://sourceforge.net/projects/ucanaccess/.

Welch, Peter D. 1983. “The Statistical Analysis of Simulation Results.” In Computer Performance Modeling Handbook, edited by Stephen Lavenberg, 267–329. New York: Academic Press.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California